CGH-Explorer

Graphical exploration and statistical analysis of array-CGH data

User Manual

October 2005

The Bioinformatics Group
Department of Informatics
University of Oslo
Norway

Department of Genetics
Institute for Cancer Research
Norwegian Radium Hospital
Norway

Section of Statistics
Department of Mathematics
University of Oslo
Norway

Contents

1. Getting started 3

2. Installation 4

3. A quick tutorial 5

4. The tool bar..... 14

5. Data import and export 15

6. Data frames. 19

7. The File menu..... 20

8. The Graph menu..... 21

9. The Tools menu..... 24

10. The Detection menu..... 25

11. The Window menu..... 28

12. The preferences menu..... 29

Appendix A..... 30

Appendix B..... 33

Appendix C..... 40

1. Getting started

CGH-Explorer is a program for visualization and statistical analysis of microarray-based comparative genomic hybridization (array-CGH) data. Some key features of the program:

- Available for most platforms (including Windows, Linux, Mac)
- User-friendly graphical user interface
- Missing value imputation, array centering and various data transformations
- Advanced graphical exploration of array-CGH data
- Identification of regions of amplification and deletion
- File export facility for plots and tables

CGH-Explorer is available as a binary executable for Windows platforms. Java source code is also available, making it easy to run the software on other platforms (this requires installation of the Java 2 platform or higher). CGH-Explorer is described in

> Lingjærde OC, Baumbusch LO, Liestøl K, Glad I, and Børresen-Dale AL (2005). CGH-Explorer: A program for analysis of CGH-data. *Bioinformatics*, 21, 821-822.

Software and documentation is available at

> http://www.ifi.uio.no/bioinf/Papers/CGH/

There are no restrictions on the use or distribution of the program. We are happy to hear about your experience with the program. Comments, ideas and suggestions for future improvements are most welcome. Please contact Dr. Ole Christian Lingjærde, Dept of Informatics, P.O. Box 1080 Blindern, N-0316 Oslo, Norway . E-mail: ole@ifi.uio.no.

2. Installation

Installation in Windows

To install the program in Windows, go to the CGH-Explorer web site

http://www.ifi.uio.no/bioinf/Papers/CGH/Software

and download the software. When you run the installer program the software will be installed in the directory "C:\Program Files\CGH-Explorer xx" where xx is the version number, and the program icon will be installed on the desktop and under *All Programs* on the *Start* menu.

To uninstall CGH-Explorer, use the Windows facility *Add or Remove Programs* located in the *Control Panel*.

Installation in other operating systems

To install CGH-Explorer in other operating systems, you need to download and compile the source code (available from the CGH-Explorer web site). To compile and run the software you also need the Java 2 standard edition development kit and runtime environment (J2SE JDK+JRE). These are available from Sun's web site (look for Desktop Java on the web site http://java.sun.com/j2se).

3. A quick tutorial

The following tutorial is designed to get you started quickly with CGH-Explorer. The main components of the program are shown, and you learn how to import a data file, perform simple preprocessing of the data, plot the data and search for copy number alterations. For a complete overview and more details on CGH-Explorer, please see later chapters. Before you start on the tutorial, download the sample data set from the CGH-Explorer download central and save it under the name "cghdata.txt".

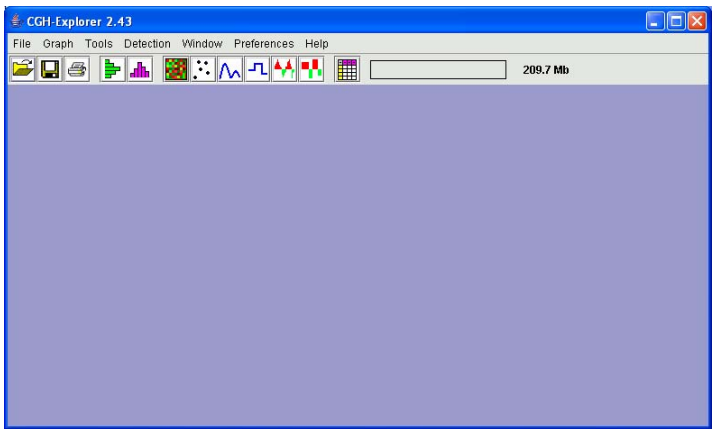
Step 1: Launching the program for the first time

When you launch the program for the first time, a file dialog appears. Use the dialog to specify the *working directory*, i.e. the directory you want to use as the default start location for file dialogs when you import or export files.

Your selection will be stored for later sessions as well. Normally, you would like to select the directory where your CGH data files typically reside. You can change the location of the working directory at any time, using the Preferences menu.

Step 2: Importing data

Having selected the working directory, the application main window appears:

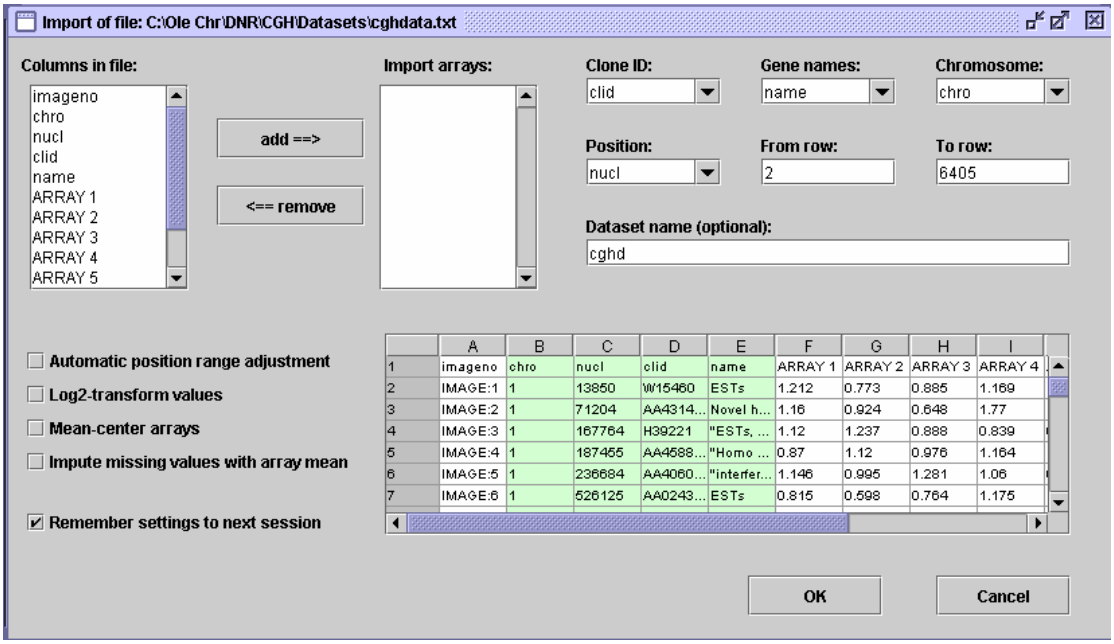


Notice the tool bar located below the menu bar. Many commonly used commands are available from the tool bar; however you should nevertheless familiarize yourself with the menus in order to make the best possible use of CGH-Explorer.

We import the sample data file cghdata.txt (available from the CGH-Explorer download central; you need to download the file to your file system before continuing) using the command *File | Import data.*. (also available leftmost on the tool bar). Locate the data file and open it.

The import function recognizes tab-delimited text files with a specific column format and with names that end with ".txt". The data file format is described in chapter 5. If your data consist of a series of files (one for each array), create a new file with columns for gene ID, gene name, chromosome number, position on chromosome measured in nucleotides, and one column of CGH data for each array. Then save the file as a tab-delimited text file. This is all easily done in e.g. Excel.

The import wizard now appears (see figure below). This dialog is used to specify how to interpret the columns in the data file. See chapter 5 for a detailed description of all entries in this dialog.



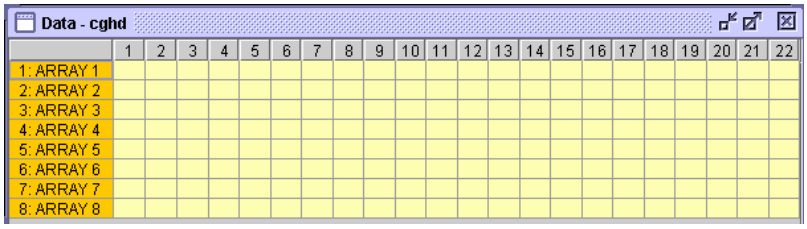
For our data set, we need to specify which columns in the file are to be treated as CGH data. Select the columns "ARRAY 1", ..., "ARRAY 8" in the list titled "Columns in file:" (click on "ARRAY 1" in the list, then hold down the shift key while you click on "ARRAY 8" in the list). Use the button **add ==>** in order to copy these list elements to the field titled "Import arrays:".

There are some actions that you commonly want to perform on the data at import, such as log-transforming or mean-centering the CGH measurements. See the check boxes in the import wizard for such optional actions. These actions are performed in the order that they appear in the dialog (from top to bottom).

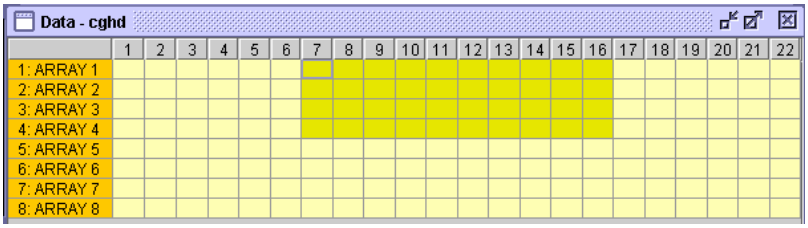
In this tutorial, we do not change the settings for the optional actions. Click "OK" to proceed.

Step 3: The data frame

The imported data set is shown in CGH-Explorer as a *data frame* (see figure below). Rows in the data frame correspond to arrays and columns correspond to chromosomes.



Plot actions in CGH-Explorer are always performed on *subsets* of a data set. To select a subset, mark a rectangular area in the data frame using the mouse. Below, we have selected the subset consisting of the first four arrays and chromosomes 7 to 16:



Immediately after import of a data file, the newly created data frame is active (you can see this by noting that the color of the title bar on the data frame is blue). Some actions (such as the creation of a plot) will render the data frame inactive. You then need to *reactivate* the data frame if you want to perform any further operations on the data set. You do this by pointing the cursor at the title bar of the data frame and clicking the left mouse button (you may point anywhere else in the data frame when you click the mouse button, but that will erase your current subset selection).

Apart from plot commands, all operations on data sets are performed on the whole data set (even if you have selected a subset). If you do need to perform one of these operations on a subset of the data, you have to prepare a new data file consisting of the relevant subset and import this file into a separate data frame.

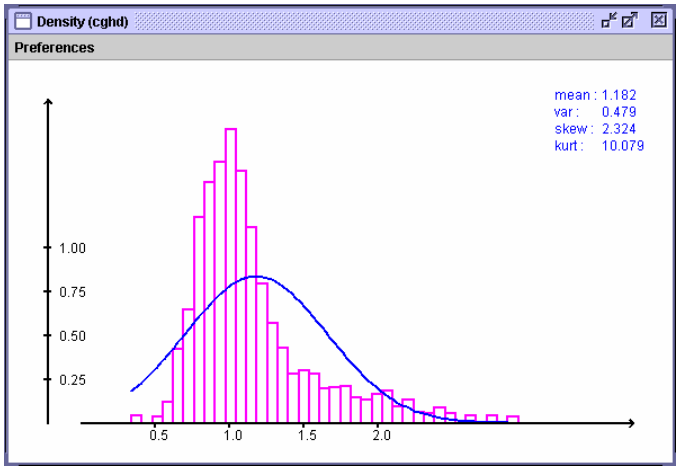
Step 4: Log-transforming the data

It is often preferable to work with log-transformed ratios rather than ratios. For example, the log-transform tends to make the variation of the values less dependent on the magnitude of the values, and the log-transform also reduces the skewness of highly skewed distributions (Amaratunga and Cabrera 2004). In addition, some of the tools in CGH-Explorer (including the detection of deletions and amplifications) assumes that the data are on logarithmic scale.

Log-transforming the data is one of the optional actions you can take during data import (see step 2). It can also be achieved after import as follows. First make sure the data frame is active. Transformations in CGH-Explorer are always applied to the whole data set, so it does not matter what the current selection of arrays and chromosomes are. Give the command *Tools* | *Transform...* and pick the transform Log2(x). Click "OK" to apply the transformation to the data.

Step 5: Plotting the empirical distribution

To see the empirical distribution of your data, first select the arrays and the chromosomes in the data frame that you want to consider. Then give the command *Graph* | *Density of CGH values.* A plot similar to this one will appear:

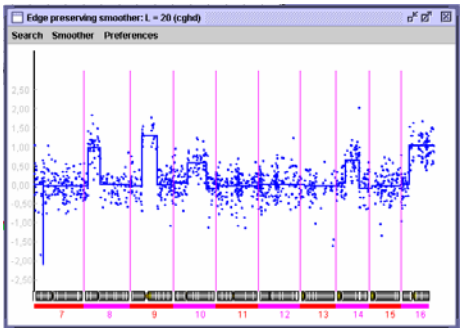


A density histogram is shown, together with a Gaussian density fitted to the data (using maximum likelihood). The mean, variance, skewness and excess kurtosis of the data are shown in the upper right corner (a normal distribution has zero skewness and zero excess kurtosis). To alter the horizontal range of the plot or the number of bins in the histogram, use the *Preferences* menu on the plot window.

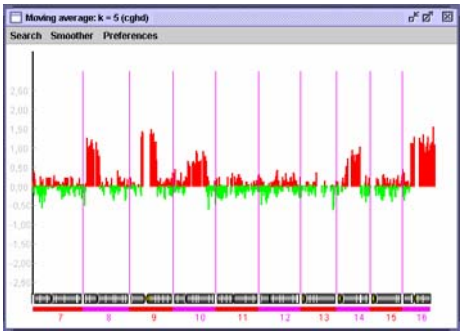
Notice that there is some skewness (i..e. a lack of symmetry in the distribution) present and also some positive kurtosis (i.e. heavier tails than for a normal distribution).

Step 6: Plotting the data

Activate the data set and select a subset of the data (e.g. chromosomes 7 to 16 and ARRAY 4). We now demonstrate two ways of visualizing the data. The first is a scatter plot of the data along with a curve fitted to the data using an edge preserving smoother. (see Appendix A for details about smoothers). To create this graph, give the command *Graph* | *Line plot* | *Edge preserving smoother*:



If you don't want to see the vertical lines in the plot that indicate the boundaries between the chromosomes, you may go into the *Preferences* menu on the graph window to turn this feature off. At the bottom of the plot you see what chromosomes are shown, as well as cytobands for each chromosome. To demonstrate another way of visualizing the data, reactivate the data frame (by clicking on the title bar of the data frame) and give the command *Graph* | *Stem plot* | *Moving average smoother*:



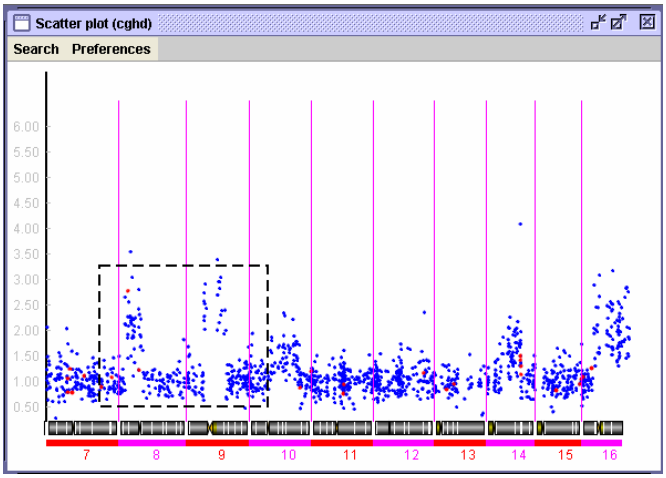
This is a stem plot that shows only a fit to the data and not the data themselves. Rather than plotting the fit as a curve, the fit is plotted as a sequence of vertical bars, one for each gene. The height of a bar is the fitted value for that gene. The fitting procedure in this case is a moving average smoother (see Appendix A for details).

Various plot options are available from the *Preferences* menu on the plot window. You can hide the vertical lines separating the chromosomes, superpose a grid of horizontal lines, and show names of cytobands.

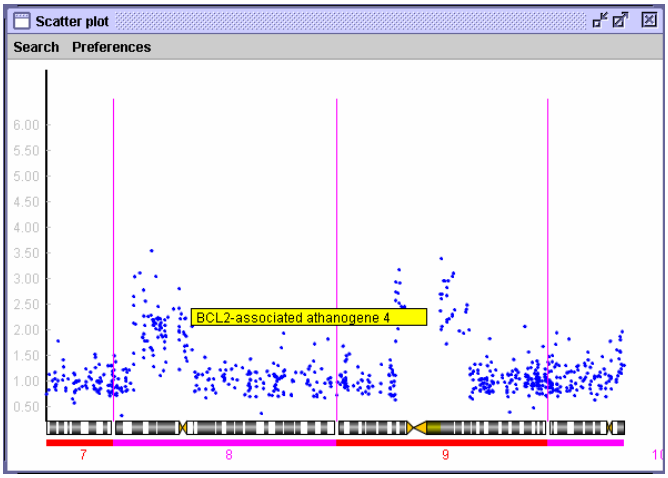
The vertical plot range, the number of tick marks, and the color and size of the points are also adjustable. The *Search* menu on the plot window allows you to search for and identify all genes that have a certain phrase in their gene name.

Step 7: Exploring details in a plot

Using the mouse you can zoom in on any part of a plot. Just point the mouse inside the window, click the left mouse button and move the mouse while keeping the mouse button down. When you release the mouse button, the resulting rectangle (see figure below) will define a *zoom region* and a new plot window appears (on top of the old one) showing only the part of the genome that contains the genes inside the zoom region. You may repeat the process as many times as you like by zooming further in on the new plots made in this way. Here is an example:

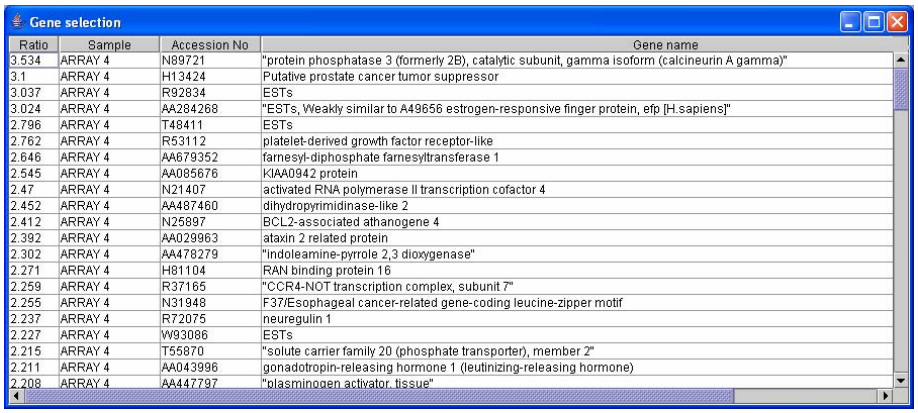


The figure below shows the result of the above zoom operation. Observe that pointing with the cursor at a dot in a scatter plot yields a yellow square containing the name of the corresponding gene/clone. Point at the ideogram at the bottom of the plot to get information about the corresponding nucleotide position.



To obtain information about all clones in a rectangular area of the plot, drag a square just as you did to zoom in, but this time press the *Ctrl* key during the operation. When you

release the mouse button, information about the clones corresponding to the dots inside the square are shown in a table:



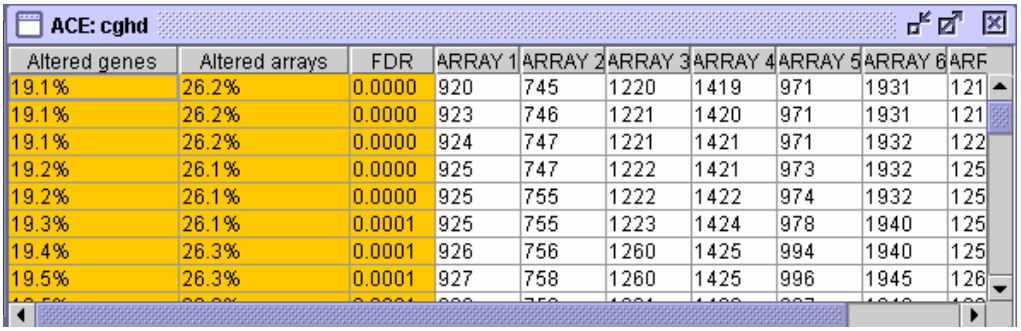
Ratio	Sample	Accession No	Gene name
3.534	ARRAY 4	N89721	"protein phosphatase 3 (formerly 2B), catalytic subunit, gamma isoform (calcineurin A gamma)"
3.1	ARRAY 4	H13424	Putative prostate cancer tumor suppressor
3.037	ARRAY 4	R92834	ESTs
3.024	ARRAY 4	AA284268	"ESTs, Weakly similar to A49656 estrogen-responsive finger protein, efp [H.sapiens]"
2.796	ARRAY 4	T48411	ESTs
2.762	ARRAY 4	R53112	platelet-derived growth factor receptor-like
2.646	ARRAY 4	AA679352	farnesyl-diphosphate farnesyltransferase 1
2.545	ARRAY 4	AA085676	KIAA0942 protein
2.47	ARRAY 4	N21407	activated RNA polymerase II transcription cofactor 4
2.452	ARRAY 4	AA487460	dihydropyrimidinase-like 2
2.412	ARRAY 4	N25897	BCL2-associated athanogene 4
2.392	ARRAY 4	AA029963	ataxin 2 related protein
2.302	ARRAY 4	AA478279	"indoleamine-pyrrole 2,3 dioxygenase"
2.271	ARRAY 4	H81104	RAN binding protein 16
2.259	ARRAY 4	R37165	"CCR4-NOT transcription complex, subunit 7"
2.255	ARRAY 4	N31948	F37/Esophageal cancer-related gene-coding leucine-zipper motif
2.237	ARRAY 4	R72075	neuregulin 1
2.227	ARRAY 4	W93086	ESTs
2.215	ARRAY 4	T55870	"solute carrier family 20 (phosphate transporter), member 2"
2.211	ARRAY 4	AA043996	gonadotropin-releasing hormone 1 (leutinizing-releasing hormone)
2.208	ARRAY 4	AA447797	"plasminogen activator, tissue"

Point the mouse at any row in the table and click the left mouse button. This initiates a search for more information about the clone in that row. By default, Internet Explorer will start up and perform a search on Entrez based on the clone id given in the third column of the table (titled *Accession No*).

The default search operation mentioned above may not be appropriate for you, in which case you can define your own search using *Preferences* | *Query definitions*.. in the menu on the main window.

Step 8: Detecting copy number alterations (CNAs)

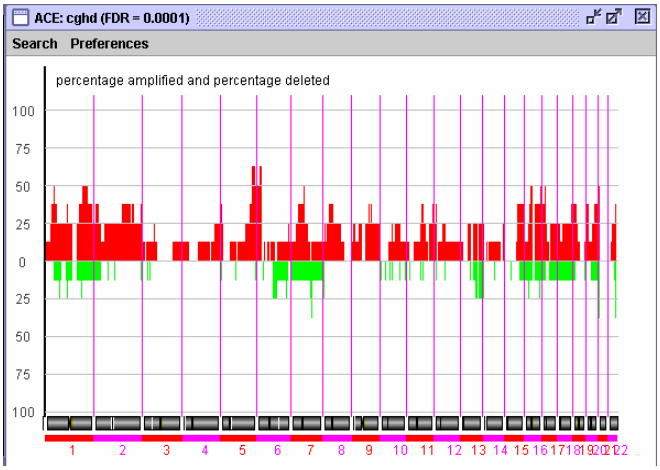
In order to detect copy number alterations (CNAs), we select the *Analysis of copy errors (ACE)* command on the *Detection* menu. After a while a table referred to as the ACE table appears:



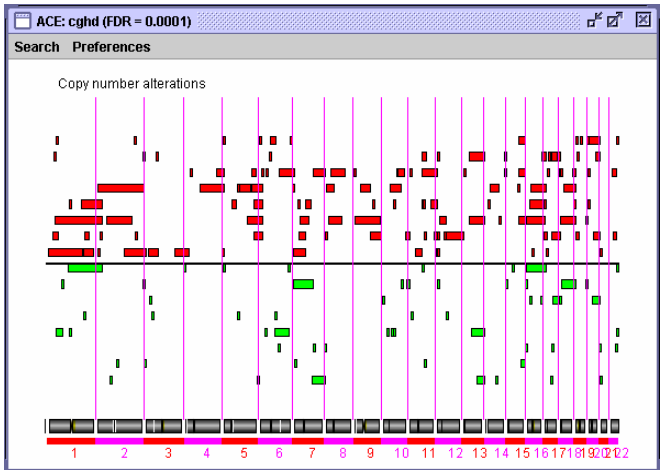
Altered genes	Altered arrays	FDR	ARRAY 1	ARRAY 2	ARRAY 3	ARRAY 4	ARRAY 5	ARRAY 6	ARF
19.1%	26.2%	0.0000	920	745	1220	1419	971	1931	121
19.1%	26.2%	0.0000	923	746	1221	1420	971	1931	121
19.1%	26.2%	0.0000	924	747	1221	1421	971	1932	122
19.2%	26.1%	0.0000	925	747	1222	1421	973	1932	125
19.2%	26.1%	0.0000	925	755	1222	1422	974	1932	125
19.3%	26.1%	0.0001	925	755	1223	1424	978	1940	125
19.4%	26.3%	0.0001	926	756	1260	1425	994	1940	125
19.5%	26.3%	0.0001	927	758	1260	1425	996	1945	126

Before you apply ACE to you data, always make sure that your data are on log-scale and that the log ratios are centered such that the expected log ratio for a normal gene is (approximately) zero. Otherwise, the ACE algorithm fails to work properly.

Each row in the ACE table corresponds to a particular trade-off between type I error (false positives) and type II error (false negatives). Select any row by pointing the cursor at it and clicking the left mouse button. A new window will then pop up, similar to this one:

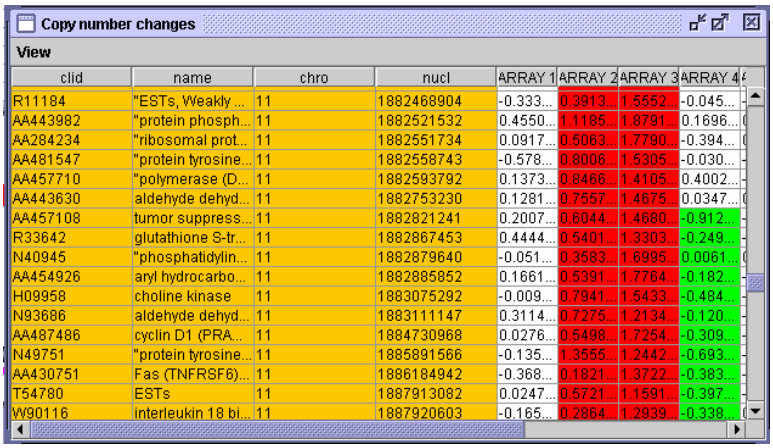


For each gene the height reflects *the proportion of arrays* for which the gene is amplified or deleted. Positive values are shown in red and indicate the proportion of arrays for which a gene is classified as amplified. Negative values are shown in green and indicate the proportion of arrays for which a particular gene is classified as deleted. Using the *Preferences* menu on the graph window you may change the plot type to *alterations (split plot)* to show instead the individual copy number alterations (CNAs) in each array:



Here, amplifications are shown in the upper part and deletions in the lower part of the plot. Each row in the upper (lower) part of the plots corresponds to one array (ordered from top to bottom).

The above graph can be converted to a tabular representation by making the graph window active and selecting the menu command *Tools* | *Convert to table.* Here is an example:



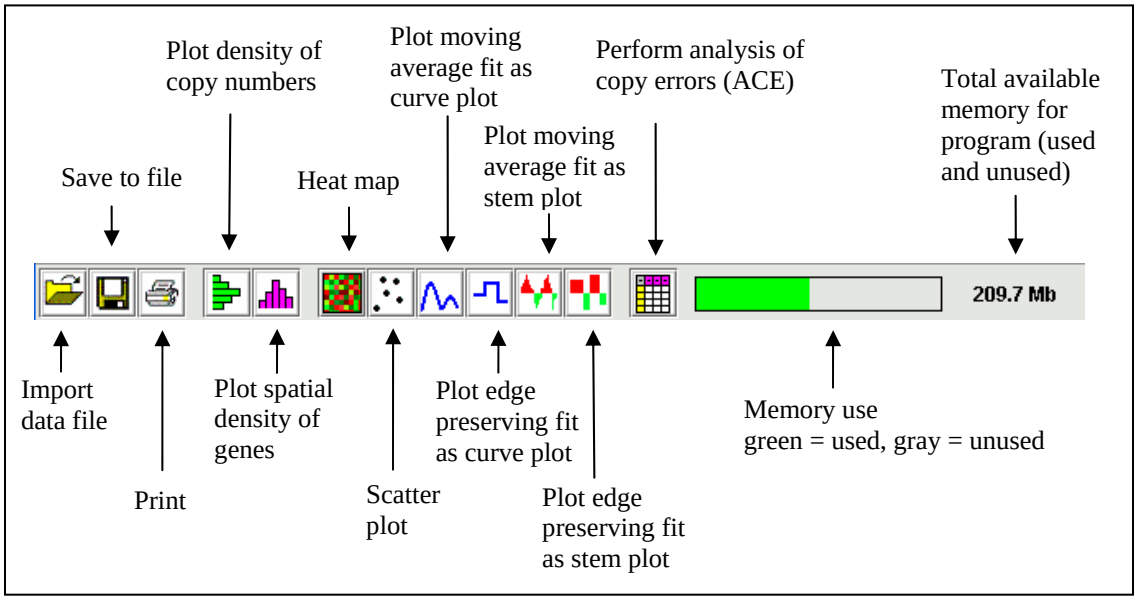
clid	name	chro	nucl	ARRAY 1	ARRAY 2	ARRAY 3	ARRAY 4
R11184	"ESTs, Weakly ...	11	1882468904	-0.333...	0.3913...	1.5552...	-0.045...
AA443982	"protein phosph...	11	1882521532	0.4550...	1.1185...	1.8791...	0.1696...
AA284234	"ribosomal prot...	11	1882551734	0.0917...	0.5063...	1.7790...	-0.394...
AA481547	"protein tyrosine...	11	1882558743	-0.578...	0.8006...	1.5305...	-0.030...
AA457710	"polymerase (D...	11	1882593792	0.1373...	0.8466...	1.4105...	0.4002...
AA443630	aldehyde dehyd...	11	1882753230	0.1281...	0.7557...	1.4675...	0.0347...
AA457108	tumor suppress...	11	1882821241	0.2007...	0.6044...	1.4680...	-0.912...
R33642	glutathione S-tr...	11	1882867453	0.4444...	0.5401...	1.3303...	-0.249...
N40945	"phosphatidylin...	11	1882879640	-0.051...	0.3583...	1.6995...	0.0061...
AA454926	aryl hydrocarbo...	11	1882885852	0.1661...	0.5391...	1.7764...	-0.182...
H09958	choline kinase	11	1883075292	-0.009...	0.7941...	1.5433...	-0.484...
N93686	aldehyde dehyd...	11	1883111147	0.3114...	0.7275...	1.2134...	-0.120...
AA487486	cyclin D1 (PRA...	11	1884730968	0.0276...	0.5498...	1.7254...	-0.309...
N49751	"protein tyrosine...	11	1885891566	-0.135...	1.3555...	1.2442...	-0.693...
AA430751	Fas (TNFRSF6)...	11	1886184942	-0.368...	0.1821...	1.3722...	-0.383...
T54780	ESTs	11	1887913082	0.0247...	0.5721...	1.1591...	-0.397...
W90116	interleukin 18 bi...	11	1887920603	-0.165...	0.2864...	1.2939...	-0.338...

Each row in the table corresponds to a particular gene. The first four columns are the clone id, the gene name, the chromosome number and the position of the gene on the chromosome.

The table above can be exported to file. The resulting file may also be imported back into CGH-Explorer as a data set in order to plot loss and gain regions for individual samples. See chapter 10 for more information.

4. The tool bar

All commands in CGH-Explorer are available from the menus. Some of the most commonly used commands are also available on the tool bar:


Plot density of copy numbers
Plot moving average fit as curve plot
Perform analysis of copy errors (ACE)
Total available memory for program (used and unused)
Save to file
Heat map
Plot moving average fit as stem plot
209.7 Mb
Import data file
Print
Plot spatial density of genes
Scatter plot
Plot edge preserving fit as curve plot
Plot edge preserving fit as stem plot
Memory use
green = used, gray = unused

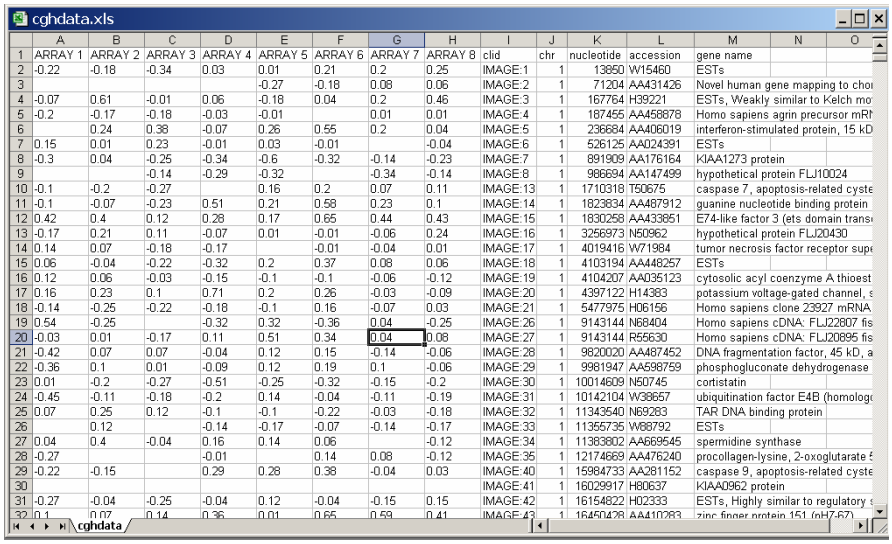
To use the tool bar, first select the target object (data frame, graph or table) of the command. For example, to print a graph, you first select the graph window (by left-clicking the mouse on the window) and then left-click on the print symbol on the tool bar. To make a plot, you have to (1) select a data set by left-clicking on a data selection window; (2) select the arrays and chromosomes that you want to plot; and (3) left-click on the wanted plot symbol on the tool bar.

5. Data import and export

CGH-Explorer allows you to import and analyze simultaneously as many data sets as you like, limited only by your computer's memory and speed.

4.1 Input data file format

To analyze a data set consisting of any number of arrays, first collect the data in a single tab-delimited text file. Here is an example of a data set (shown in Excel):



cghdata.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ARRAY 1	ARRAY 2	ARRAY 3	ARRAY 4	ARRAY 5	ARRAY 6	ARRAY 7	ARRAY 8	clid	chr	nucleotide	accession	gene name
2	-0.22	-0.18	-0.34	0.03	0.01	0.21	0.2	0.25	IMAGE:1	1	13850	W15460	ESTs
3					-0.27	-0.18	0.08	0.06	IMAGE:2	1	71204	AA431426	Novel human gene mapping to chor
4	-0.07	0.61	-0.01	0.06	-0.18	0.04	0.2	0.46	IMAGE:3	1	167764	H39221	ESTs, Weakly similar to Kelch mo
5	-0.2	-0.17	-0.18	-0.03	-0.01		0.01	0.01	IMAGE:4	1	187455	AA458878	Homo sapiens agrin precursor mRN
6		0.24	0.38	-0.07	0.26	0.55	0.2	0.04	IMAGE:5	1	236684	AA406019	interferon-stimulated protein, 15 kD
7	0.15	0.01	0.23	-0.01	0.03	-0.01		-0.04	IMAGE:6	1	526125	AA024391	ESTs
8	-0.3	0.04	-0.25	-0.34	-0.6	-0.32	-0.14	-0.23	IMAGE:7	1	891909	AA176164	KIAA1273 protein
9			-0.14	-0.29	-0.32		-0.34	-0.14	IMAGE:8	1	986694	AA147499	hypothetical protein FLJ10024
10	-0.1	-0.2	-0.27		0.16	0.2	0.07	0.11	IMAGE:13	1	1710318	T50675	caspase 7, apoptosis-related cyste
11	-0.1	-0.07	-0.23	0.51	0.21	0.58	0.23	0.1	IMAGE:14	1	1823834	AA487912	guanine nucleotide binding protein
12	0.42	0.4	0.12	0.28	0.17	0.65	0.44	0.43	IMAGE:15	1	1830258	AA433851	E74-like factor 3 (ets domain transc
13	-0.17	0.21	0.11	-0.07	0.01	-0.01	-0.06	0.24	IMAGE:16	1	3256973	N50962	hypothetical protein FLJ20430
14	0.14	0.07	-0.18	-0.17		-0.01	-0.04	0.01	IMAGE:17	1	4019416	W71984	tumor necrosis factor receptor supe
15	0.06	-0.04	-0.22	-0.32	0.2	0.37	0.08	0.06	IMAGE:18	1	4103194	AA448257	ESTs
16	0.12	0.06	-0.03	-0.15	-0.1	-0.1	-0.06	-0.12	IMAGE:19	1	4104207	AA035123	cytosolic acyl coenzyme A thioest
17	0.16	0.23	0.1	0.71	0.2	0.26	-0.03	-0.09	IMAGE:20	1	4397122	H14383	potassium voltage-gated channel, s
18	-0.14	-0.25	-0.22	-0.18	-0.1	0.16	-0.07	0.03	IMAGE:21	1	5477975	H06156	Homo sapiens clone 23927 mRNA
19	0.54	-0.25		-0.32	0.32	-0.36	0.04	-0.25	IMAGE:26	1	9143144	N68404	Homo sapiens cDNA: FLJ22807 fis
20	-0.03	0.01	-0.17	0.11	0.51	0.34	0.04	0.08	IMAGE:27	1	9143144	R55630	Homo sapiens cDNA: FLJ20895 fis
21	-0.42	0.07	0.07	-0.04	0.12	0.15	-0.14	-0.06	IMAGE:28	1	9820020	AA487452	DNA fragmentation factor, 45 kD, a
22	-0.36	0.1	0.01	-0.09	0.12	0.19	0.1	-0.06	IMAGE:29	1	9981947	AA598759	phosphogluconate dehydrogenase
23	0.01	-0.2	-0.27	-0.51	-0.25	-0.32	-0.15	-0.2	IMAGE:30	1	10014609	N50745	cortistatin
24	-0.45	-0.11	-0.18	-0.2	0.14	-0.04	-0.11	-0.19	IMAGE:31	1	10142104	W38657	ubiquitination factor E4B (homologo
25	0.07	0.25	0.12	-0.1	-0.1	-0.22	-0.03	-0.18	IMAGE:32	1	11343540	N69283	TAR DNA binding protein
26		0.12		-0.14	-0.17	-0.07	-0.14	-0.17	IMAGE:33	1	11355735	W88792	ESTs
27	0.04	0.4	-0.04	0.16	0.14	0.06		-0.12	IMAGE:34	1	11383802	AA669545	spermidine synthase
28	-0.27			-0.01		0.14	0.08	-0.12	IMAGE:35	1	12174669	AA476240	procollagen-lysine, 2-oxoglutarate 5
29	-0.22	-0.15		0.29	0.28	0.38	-0.04	0.03	IMAGE:40	1	15984733	AA281152	caspase 9, apoptosis-related cyste
30									IMAGE:41	1	16029917	H80637	KIAA0962 protein
31	-0.27	-0.04	-0.25	-0.04	0.12	-0.04	-0.15	0.15	IMAGE:42	1	16154822	H02333	ESTs, Highly similar to regulatory s
32	0.1	0.07	0.14	0.36	0.01	0.65	0.59	0.41	IMAGE:43	1	16450428	AA410283	zinc finger protein 151 (pH7-67)

cghdata

Data files should have a title row followed by a row for each gene. There should be one column for each array in the experiment, as well as columns providing positional and other information for each gene. The file should contain at least the following columns (in arbitrary order):

- A column for each sample (array) to be analysed, giving the copy number ratios (or some transformation of the ratios, such as the logarithm of the copy number ratios) for all genes on the array. Empty cells are treated as missing values and will be imputed during import (see below for details).
- A clone/gene identifier, e.g. an accession number. It need not be unique.
- The chromosome (1,2,...,X,Y) where the gene is located (in normal DNA).
- The gene position. Positions should be given as the number of nucleotides from the start of the chromosome.
- The gene name.

Rows in the data file should be ordered increasingly with respect to chromosome number (1,2,....,X,Y) and within each chromosome increasingly with respect to gene position. *CGH-Explorer does not check this – make sure to do so yourself before importing a file.*

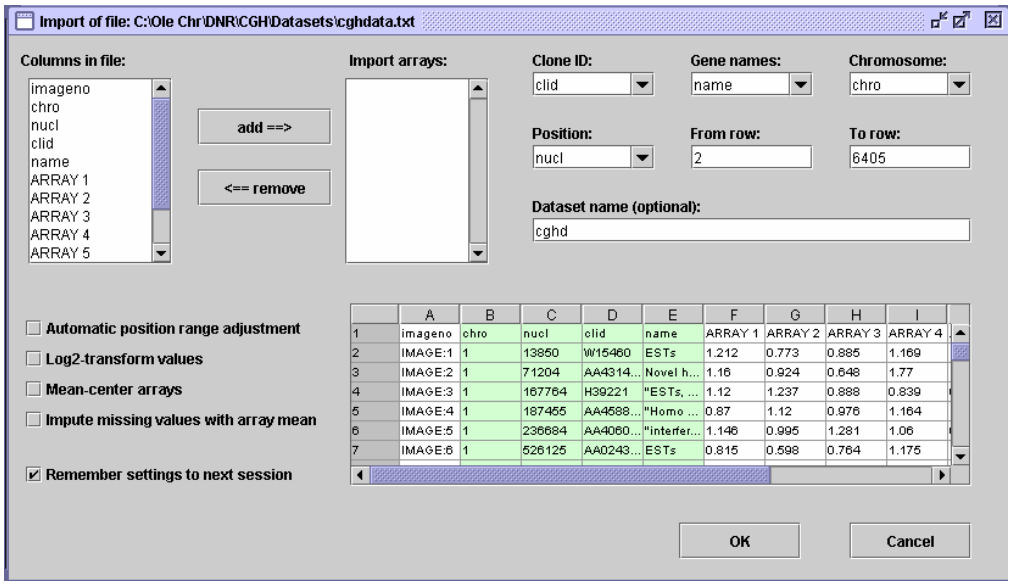
4.2 Missing values in input data

Blank fields in the data file are regarded as missing values. At import, missing values are automatically filled in using one of these imputation methods:

- Imputation with zeros
- Imputation with array mean (i.e. impute with the average of all values on an array).

4.3 The import wizard

Suppose you have downloaded the sample data set cghdata.txt from the program's web site. Choose *File | Import Data..,* pick the file cghdata.txt and click "OK". The *import wizard* will then appear:



This dialog is used to specify

- which columns in the file to import, and how to interpret them
- optional operations to be performed on the data during import

The various components of the import wizard dialog are now described.

4.3.1 The preview table

A condensed view of the data file – a preview table - is shown in the lower right corner of the import wizard. Notice that some columns are shaded in green. The shaded columns are those that are selected for import so far. The preview table serves two purposes: it tells you how CGH-Explorer splits the file into columns (this may be useful to detect errors in the file format) and it tells you which columns have been selected for import.

4.3.2 Column specification

Notice that four columns have already been selected for import. CGH-Explorer has recognized the titles given at the top of these columns (chro, nucl, clid and name) and knows what function to assign to these columns. For example, a column with the title 'chro' will be interpreted by CGH-Explorer as a column with gene identifiers. If you want to use another column as your gene identifier, select another column in the selection box **Clone ID** in the dialog.

The table below gives an overview of entries in the dialog that are related to the selection of columns. The first entry in the table (**Import arrays**) allows several columns to be selected, whereas other entries allow only one column to be selected.

Dialog entry	Treat column(s) as...	Automatic recognition
Import arrays	CGH data	
Clone ID	gene ID	if column title is 'clid'
Gene names	gene name	if column title is 'name'
Chromosome	chromosome number (1,2,....,X,Y)	if column title is 'chro'
Position	nucleotide position on chromosome	if column title is 'nucl'
From row	first row in file to import	
To row	last row in file to import	
Dataset name	optional name to use in program	

To specify which arrays to analyze, select one or several columns in the left hand list (to select two non-adjacent elements, hold the Ctrl-key down while clicking on elements in the list; to select two or more adjacent elements, select the first, then select the last while holding down the shift key). Then click the button **add** ==> in order to copy these elements to the field **Import arrays**. Use **remove** ==> in a similar manner to remove columns from your current selection.

4.3.3 Optional operations to be performed during import

The import dialog also contains five check boxes; see table below for a description.

Check box	Description
Automatic position range adjustment	Scale nucleotide positions if necessary
Log2-transform values	Apply logarithmic transformation to CGH values
Mean-center arrays	For each array, subtract mean from all CGH values
Impute missing values with array mean	Replace missing values by array mean
Remember settings to next session	Save import wizard settings to future sessions

Note that if you don't specify that you want to impute missing values with array mean, then CGH-Explorer will simply replace missing values by zeros.

When you have filled in the empty fields in the import dialog window and clicks "OK", the data will be imported. The program also produces a new file on the same directory with the same name as the data file and ending with ".cgh". This file contains the information you submitted in the import dialog window, and the next time you open the data file, all fields will be filled in (but you still have the option to make changes, resulting in an update of the .cgh file).

A comment regarding the option to perform automatic position range adjustment: in some cases (such as for the sample data on the web site), nucleotide positions may be inaccurate and even extend beyond the actual range for a given chromosome. This may happen, for example, if nucleotide positions are obtained on the basis of an early draft of the human genome. In such cases, you may want to perform automatic position adjustment by scaling the position values to fit the length of the chromosome exactly. This will only give a crude approximation to the correction positions, and results obtained after such rescaling of the positions of genes on a chromosome should be interpreted with that in mind.

4.4 Saving data tables to file

To save a table made in CGH-Explorer as a tab-delimited file, you have two options:

- Select the table (i.e. activate the window by pointing at it and clicking the left mouse button) and choose *File* | *Save As.*.
- You may also save a table (or part of it) by selecting all (or some) elements in the table, making a copy of the selection by typing *Ctrl-C*, and then pasting the result into an Excel sheet.

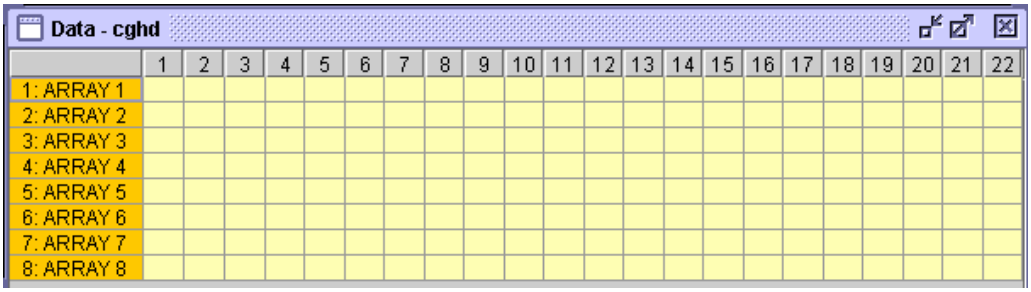
4. 5 Saving graphics to file

Plots produced by CGH-Explorer can be printed or saved. To save a plot, first make the graph window active. You now have three options:

- Choose *File* | *Save As.*. to save the plot as a Postscript file.
- Choose *File* | *Print Graph.*. and select a printer that gives you the option to print to file. Check the box 'Print to file' and select 'Print'. The file (with the ending ".prn") is essentially a postscript file and should be treated as such by postscript printers. If you have installed Adobe Acrobat PDFMaker, you may also be able to print the document to 'Adobe PDF' to produce a pdf-file.
- Choose *Tools* | *Convert to table* to convert the graph to a numerical representation in a table. This table may then saved as described in the preceding section.

6. Data frames

An imported data set is initially shown as an empty table with one column for each chromosome and one row for each array, referred to as a *data frame*:



A data frame serves two purposes:

- All operations on a data set will be performed on the currently active data frame. To make a data frame active, point at the title bar and click the left mouse button.

- Prior to some operations (such as making a plot of the data), the user is required to specify what part of a data set to use. This is done by using the mouse to select one or several of the boxes (each corresponding to a particular chromosome and a particular array) in a data frame.

In the current version of the program, selection of multiple boxes in a data frame is restricted to selecting a rectangular region in the data frame.

You may temporarily close a data frame by left-clicking on the symbol in the upper right corner of the window. The data set will still be available in CGH-Explorer; use the *Window* menu to reopen the window.

7. The File menu

Import data..

Import a data file. Data files are text files consisting of copy number data and other information for a number of genes (or clones) and for one or more arrays. See chapter 5 for a description of the structure of data files and the various options related to import of such files into CGH-Explorer.

The file dialog that appears during import always defaults to show the working directory. To change the working directory, use the *Preferences* menu.

Save As..

Save a table or a graph to file. Tables are saved as tab-delimited text files, whereas graphs are saved as postscript files. The file dialog that appears always defaults to show the working directory. To change the working directory, use the *Preferences* menu.

Page Setup..

Set print preferences.

Print Preview

Show a print preview on screen.

Print Graph..

Print a graph.

Exit

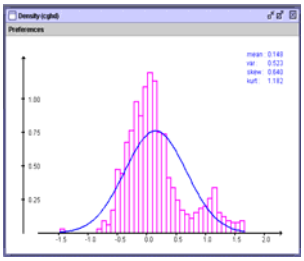
Exit the program. Note that all graphs and other results are lost unless you save them before you exit.

8. The Graph menu

Use the graph menu to visualize (a subset of) your data. First make the appropriate data window active (by pointing the mouse at the title bar of the data window and clicking the left mouse button). Then select the desired subset of arrays and chromosomes in the data window. Finally, select one of the commands on the graph menu (most of them are also available on the tool bar).

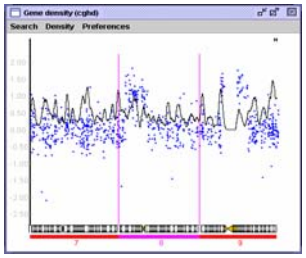
6.1 Description of commands

Density of CGH values



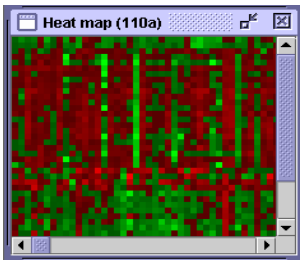
Plot the distribution of the CGH values as a histogram. The default horizontal range of the plot is [a, b] where a is the 0.5% quantile of the data and b is the 99.5% quantile of the data. Using the *Preferences* menu on the graph window you can change the horizontal range and the number of classes (bins) in the histogram. The graph also shows a normal density fitted by maximum likelihood to the subset of the data that is inside the interval [a, b] above. In the upper right corner of the plot the mean, variance, skewness and excess kurtosis of the data are shown (for a standard normal distribution, these have values 0, 1, 0, 0 respectively).

Density of genes



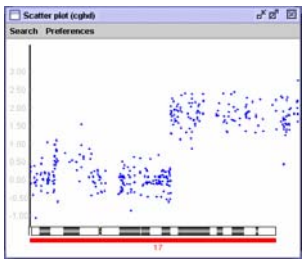
Plot and estimate of the spatial distribution of the genes, using an Epanechnikov kernel density estimator. Use the *Density* menu on the graph window to change the smoothness of the estimate. See *scatter plot* for a description of the other menus on the graph window. Some details: on a chromosome, let x_k be the gene positions and define a uniform grid. In each grid point t, the plotted value is proportional to $\hat{f}(t) = \sum r(\sqrt{5}(t - x_k)/w)$, where $r(t) = \frac{3}{4}(1 - t^2/5)/\sqrt{5}$ for $t^2 < 5$ and $r(t) = 0$ otherwise. The smoothness of the estimated curve is determined by the size w of the neighborhood. By default, $w = 4Mb$ and the total number of grid points on the genome is 5000.

Heat map



Plot a heat meap of the CGH values. The rows in the heat map correspond to arrays (ordered from top to bottom) and the columns correspond to genes (ordered from left to right). Green squares correspond to negative values and red squares correspond to positive values.

Scatter plot



Show a scatter plot (point plot) of the data.

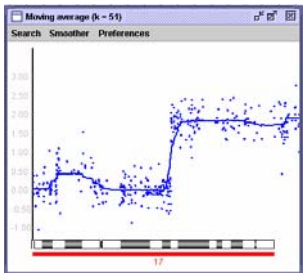
The *Search* menu consists of two commands:

Find gene... Search for genes with names containing a particular phrase. Use to identify all points for which the gene name *contains* a word such as "transcription" by typing the phrase "transcription" without the apostrophes in the search dialog. Use to identify all points for which the gene name *is equal to* a particular phrase such as "EST", by using the phrase "EST" with the double apostrophes.
Clear search result. Restore the original plot.

The *Preferences* menu is used to change the appearance of the plot. Menu entries are:

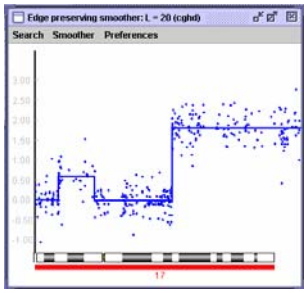
Chromosome separators. Show/hide vertical lines separating the chromosomes in plots that span several chromosomes.
Grid. Show/hide horizontal guiding lines positioned at the same height as the tick marks on the vertical axis.
Cytoband ID. Show/hide names of the cytobands. When this feature is on, the names appear at the bottom of the plot above the ideogram.
Tick marks. Adjust the spacing between the tick marks on the vertical axis, and the spacing between the grid lines if grid is turned on.
Grid type. Choose between normal (equispaced) tick marks/grid and exponentially spaced tick marks/grid in which the grid lines are positioned at 1x, 2x, 4x,....
Vertical range. Adjust the vertical range of the plot.
Plot preferences. Change the size, shape and color of points and the type, width and color of lines.

Line plot | moving average smoother



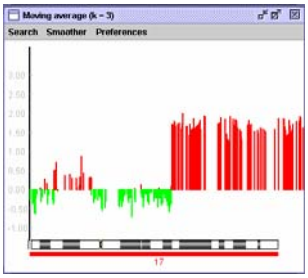
Show a line plot of the data. A line plot is a smooth representation of the data. It is useful for visual determination of systematic alterations in the magnitude of the copy number ratios. See Appendix A for a description of the moving average smoother. Use the *Smoother* menu on the graph window to alter the smoothness of the estimate. See *scatter plot* for a description of the other menus on the graph window.

Line plot | edge preserving smoother



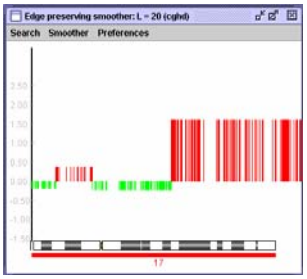
Show a line plot of the data. A line plot is a smooth representation of the data. It is useful for visual determination of systematic alterations in the magnitude of the copy number ratios. See Appendix A for a description of the edge preserving smoother. Use the *Smoother* menu on the graph window to alter the smoothness of the estimate. See *scatter plot* for a description of the other menus on the graph window.

Stem plot | moving average smoother



Show a stem plot of the data. A stem plot is a smooth representation of the data and is similar to a line plot, except that smoothed values are represented as vertical bars (red for positive values and green for negative values) rather than points joined by line segments. See Appendix A for a description of the moving average smoother. Use the *Smoother* menu on the graph window to alter the smoothness of the estimate. See *scatter plot* for a description of the other menus on the graph window.

Stem plot | edge preserving smoother



Show a stem plot of the data. A stem plot is a smooth representation of the data and is similar to a line plot, except that smoothed values are represented as vertical bars (red for positive values and green for negative values) rather than points joined by line segments. See Appendix A for a description of the moving average smoother. Use the *Smoother* menu on the graph window to alter the smoothness of the estimate. See *scatter plot* for a description of the other menus on the graph window.

9. The Tools menu

These operations can be performed on a data set (the first can also be applied to a graph).

Convert to table

Convert a data set or a graph to a numerical table.

Duplicate

Make an internal copy of a data set. This may be useful if you want to apply transformations to a data set and still want the original data set to be available for analysis.

Impute...

Impute missing values. Missing values are always imputed when you import the data into CGH-Explorer. However, CGH-Explorer keeps track of imputed values and allows you to reimpute at a later time (for example after you have applied a transformation to the data). Two imputation methods are available: impute with zeros or impute with array means.

Transform...

Apply a transformation to all the CGH values in a data set. The following transformations are available:

Transform	Description
x * a	Multiply by a constant
x / a	Divide by a constant
x + a	Add a constant
Log2(x)	Log-transform in base 2
Log2(a+x)	Add a constant and log-transform in base 2
Log10(x)	Log-transform in base 10
Log10(a + x)	Add a constant and log-transform in base 10
Sqrt(x)	Square root-transform
Sqrt(a + x)	Add a constant and square root-transform
Exp2(x)	Exponentiate in base 2
Exp10(x)	Exponentiate in base 10
Square(x)	Square-transform
Power(x, a)	Compute power transform with constant exponent
Winsorize(x, a)	Replace all x<-a with x=-a and all x>a with x=a

Mean-center genes

This replaces the CGH values $x_1,\ldots,x_n$ for a gene by new values $x_1-a,\ldots,x_n-a$ where a is the arithmetic average of $x_1,\ldots,x_n$.

Mean-center arrays

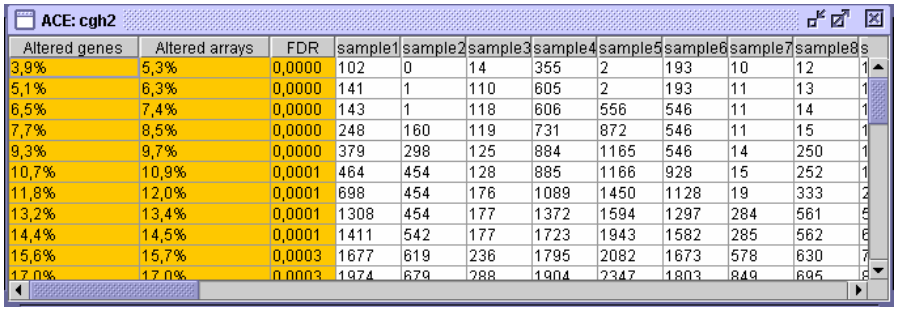
This replaces the CGH values $x_1,\ldots,x_n$ for an array by new values $x_1-a,\ldots,x_n-a$ where a is the arithmetic average of $x_1,\ldots,x_n$.

10. The Detection menu

This menu contains a single command *Analysis of copy errors (ACE).* Use this command to detect amplifications and deletions. The analysis is always performed on a whole data set; marking a selection of arrays and chromosomes in the data window has no effect.

Step 1 – computing the ACE table

The first step of the analysis is to select a data set (by pointing the cursor at the title bar of the corresponding data window and clicking the left mouse button) and performing the *Analysis of copy errors (ACE)* command on the *Detection* menu. *Note: before doing this make sure that (1) the data are on log-scale; and (2) the data are centered such that the expected value for a normal gene is zero. The latter can often be approximately achieved by array-centering the log-transformed data.* After a while a table (called the ACE table) appears:


ACE: cgh2

Altered genes	Altered arrays	FDR	sample1	sample2	sample3	sample4	sample5	sample6	sample7	sample8	s
3,9%	5,3%	0,0000	102	0	14	355	2	193	10	12	1
5,1%	6,3%	0,0000	141	1	110	605	2	193	11	13	1
6,5%	7,4%	0,0000	143	1	118	606	556	546	11	14	1
7,7%	8,5%	0,0000	248	160	119	731	872	546	11	15	1
9,3%	9,7%	0,0000	379	298	125	884	1165	546	14	250	1
10,7%	10,9%	0,0001	464	454	128	885	1166	928	15	252	1
11,8%	12,0%	0,0001	698	454	176	1089	1450	1128	19	333	2
13,2%	13,4%	0,0001	1308	454	177	1372	1594	1297	284	561	5
14,4%	14,5%	0,0001	1411	542	177	1723	1943	1582	285	562	6
15,6%	15,7%	0,0003	1677	619	236	1795	2082	1673	578	630	7
17,0%	17,0%	0,0003	1974	679	288	1904	2347	1803	849	695	8

Rows in the ACE table correspond to different trade-offs between two conflicting interests: keeping the number of false positives low (low type I error) and keeping the number of false negatives low (low type II error). The columns in the table are:

- Altered genes: the number of called (i.e. significant) genes on an array, averaged over all arrays. This is the total number of significant features in all arrays divided by the number of arrays divided by the number of genes multiplied by 100. For example, with 10 arrays and 1000 features on the array, a total of 1500 significant features in all arrays would give 15% altered genes.

- Altered arrays: the number of arrays displaying a particular gene aberration, averaged over all aberrant genes (i.e. genes that are called significant for at least one array). This is the total number of significant features in all arrays divided by the number of arrays divided by the number of aberrant genes multiplied by 100. No distinction is made here between type of aberration, i.e. an aberration is either an amplification or a deletion.

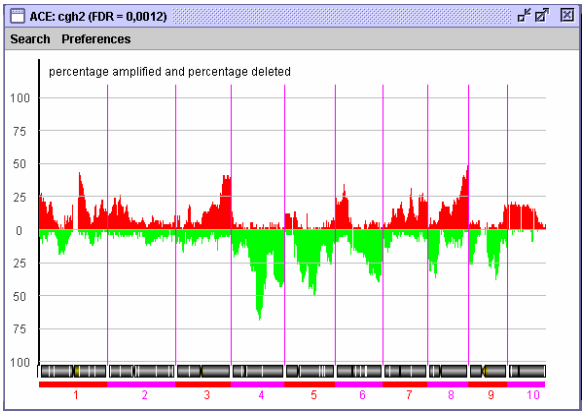
- FDR: the positive false discovery rate (pFDR). This is an estimate of the ratio between the expected number of false positives and the number of positives. See Appendix B for details.

- The remaining columns show, for each array, the number of significant features.

Step 2 – selecting the number of significant features

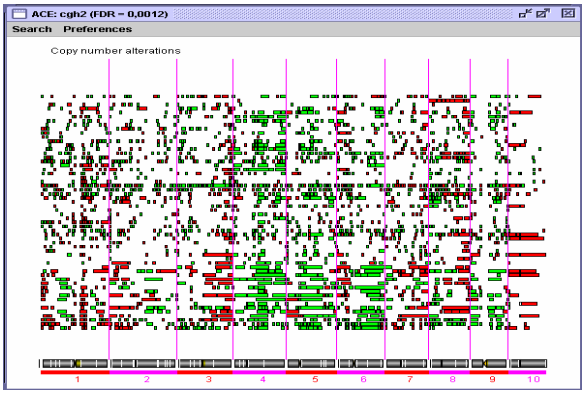
You select your preferred trade-off between type I error and type II error by clicking on one of the rows in the ACE table. A graph of the results for that cut-off point will then be shown in a separate window. You can choose between three different plot types (the first one shown below is the default; use the Preferences menu on the plot window to change to one of the others):

1) frequencies:



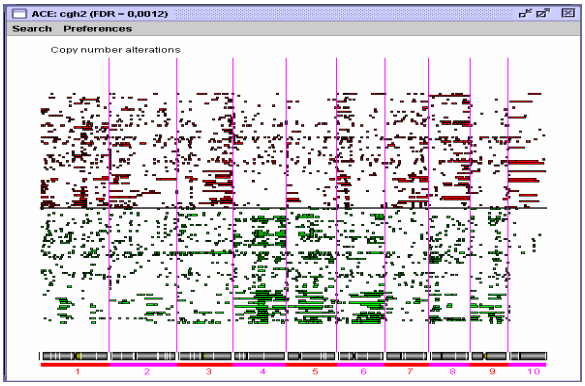
For each gene, the percentage of arrays with amplification of that gene is shown in red and the percentage of arrays with deletion of that gene is shown in green.

2) alterations (single plot):



Each copy number alteration is shown as an interval (red for amplifications and green for deletions). There is one row for each array (ordered from top to bottom).

3) alterations (split plot):

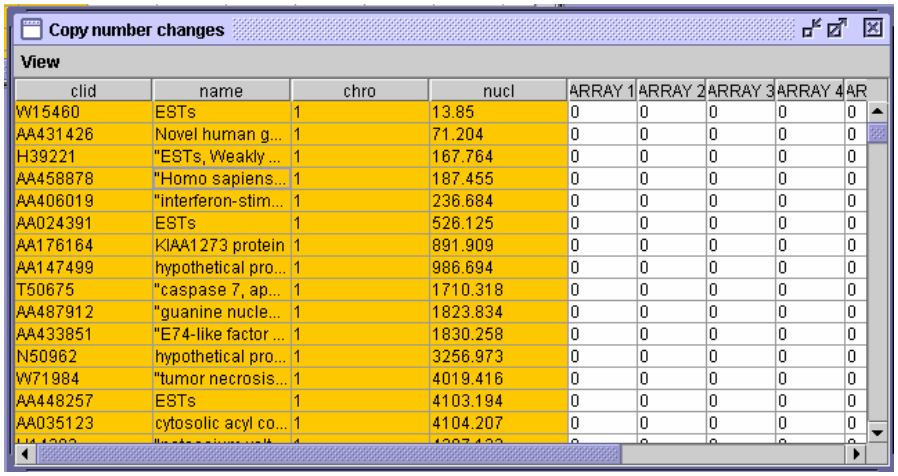


Each copy number alteration is shown as an interval (red for amplifications and green for deletions). There is one row for each array (ordered from top to bottom). Amplifications and deletions are shown separately.

Use *Preferences* in order to change the appearance of the above plots, e.g. to change the color scheme in the alterations plots above. You may zoom in on a particular part of the plot, as described earlier in the section on graphical exploration.

Step 3 – converting to a numerical table

The graph above can be converted to a tabular representation by making the graph window active and giving the menu command *Tools* | *Convert to table.* Here is an example of such a table:



clid	name	chro	nucl	ARRAY 1	ARRAY 2	ARRAY 3	ARRAY 4	AR
W15460	ESTs	1	13.85	0	0	0	0	0
AA431426	Novel human g...	1	71.204	0	0	0	0	0
H39221	"ESTs, Weakly ...	1	167.764	0	0	0	0	0
AA458878	"Homo sapiens...	1	187.455	0	0	0	0	0
AA406019	"interferon-stim...	1	236.684	0	0	0	0	0
AA024391	ESTs	1	526.125	0	0	0	0	0
AA176164	KIAA1273 protein	1	891.909	0	0	0	0	0
AA147499	hypothetical pro...	1	986.694	0	0	0	0	0
T50675	"caspase 7, ap...	1	1710.318	0	0	0	0	0
AA487912	"guanine nucle...	1	1823.834	0	0	0	0	0
AA433851	"E74-like factor ...	1	1830.258	0	0	0	0	0
N50962	hypothetical pro...	1	3256.973	0	0	0	0	0
W71984	"tumor necrosis...	1	4019.416	0	0	0	0	0
AA448257	ESTs	1	4103.194	0	0	0	0	0
AA035123	cytosolic acyl co...	1	4104.207	0	0	0	0	0

Each row in the table corresponds to a particular gene. The first four columns are the clone id, the gene name, the chromosome number and the nucleotide position of the gene. By default, the remaining columns (one for each array) show a numerical code indicating the status of each gene in each array (-1 for deletion, 0 for normal and 1 for amplification). Red cells indicate array features that are classified as amplified, while green cells indicate array features that are classified as deleted. Use the menu command *View* | *Values in table* to replace the -1/0/1 status value for each gene by the log copy number ratios.

The table include information about all genes in the data set (significant as well as non-significant ones). By default, all genes are highlighted in yellow. To highlight only those genes that are significant in at least k samples, use the menu command *View* | *Select genes.*. You may later return to highlighting all genes by using the menu command *View* | *Select genes.*. and clicking the *Cancel* button.

Step 4 – saving the table to file

Use the main menu command *File* | *Save as.*. to save the table as a tab delimited text file. Note that only the rows (genes) highlighted in yellow are saved to file.

11. The Window menu

The following operations can be performed on a data set (all these operations are available through the **Tools** menu):

Datasets

Show the data as a table of numerical values.

Tables

Make an internal copy of a data set. This may be useful if you want to apply transformations to a data set and still want the original data set to be available for analysis.

Plots

Impute missing values. Missing values are always imputed when you import the data into CGH-Explorer. However, CGH-Explorer keeps track of which CGH values have been imputed and allows you to perform imputation again at a later time.

12. The Preferences menu

The following operations can be performed on a data set (all these operations are available through the **Tools** menu):

Working directory..

Specify a new working directory, i.e. a new directory to use as the starting point in all file dialogs.

Query definitions..

Modify the query definitions.

Plot resolution

Choose between enabling and disabling the low resolution mode. In low resolution mode, graphs plot only a subset of the data points when there are many points in a plot (this may save considerable time).

Genome selection

Choose between different organisms. This affects CGH-Explorer's expectation about the number and size of chromosomes, as well as the cytoband information given in plots.

Look and feel

This potentially affects the appearance of certain dialogs and windows in CGH-Explorer. You may choose between the cross platform look and feel (the default) and the native look and feel. For example, using Windows, the native look and feel will result in Windows-looking file dialogs. Note that changes in the look and feel in the middle of a session may give some unexpected results. As a rule, change the look and feel immediately after starting a session, or at the end of the session (this will affect the next session). On some computers, a change in the look and feel may not result in any change of the appearance.

Appendix A

Smoothing

For any given array and chromosome, we have a number of pairs (x_i, y_i), $i = 1, \ldots, m$ where x_i is the position of the *i*th gene and y_i is the corresponding CGH measurement (assumed here to be given as a log-ratio). Assume that the pairs have been ordered from left to right on the chromosome, i.e. such that $x_1 \leq \cdots \leq x_m$.

Each CGH measurement y_i is determined not only by the actual copy number ratio for that gene, but also by a number of other factors, including gene-dependent and microarray-dependent factors. A simple, but useful, model for the CGH measurements is

$$y_i = f_i + noise_i, \quad i = 1, 2, \ldots, m$$

where f_i is the actual log copy number ratio for the *i*th gene and $noise_i$ is the combined contribution of all other factors. After appropriate normalization and centering of the data the noise terms $noise_i$, $i = 1, \ldots, m$ should be approximately independently and identically distributed.

Since our main interest is in the actual log ratios and not the measured log ratios, we would like to estimate the two components f_i and $noise_i$ from their sum $y_i = f_i + noise_i$. In order to do this, we make an important assumption (here stated informally): that the sequence of actual log ratios f_i, $i = 1, \ldots, m$ fluctuates much slower than the corresponding sequence of measured log ratios y_i, $i = 1, \ldots, m$. This is reasonable to assume if segments of normal DNA (for which $f_i \equiv 0$) and segments of altered DNA (in which either $f_i > 0$ or $f_i < 0$ holds for all genes in the segment) typically include several genes.

Much stronger assumptions than above are sometimes made about the data generating process. This leads to various types of models such as Markov chain models and ANOVA models. Precise estimates of the wanted quantities (such as the f_i) and their uncertainty may be obtained under such assumptions, but the quality of the results may depend heavily on the validity of the model assumptions. Although often very useful,

such procedures should only be applied after careful consideration of how well the data meet the stated assumptions. A useful alternative, particularly for early visual inspection of the data, is to apply a smoothing procedure. Smoothing procedures make only weak assumption about the data generating process (basically that the actual log ratios fluctuates slower than the measured log ratios); thus they are flexible and less prone to model misspecification. On the other hand, smoothing procedures should not be expected to produce very precise estimates of the wanted quantities, and they are generally most useful for the purpose of visualizing the data.

CGH-Explorer implements two different smoothers for use in visualization of array-CGH data: the *moving average smoother* and the *edge preserving smoother*. Whereas the former produces estimated log ratios $\hat{f}_i$ that typically vary from gene to gene (although less than the original measured log ratios), the latter produces estimated log ratios that stay constant over regions.

The moving average smoother

As before, we have a number of pairs (x_i, y_i), $i = 1,\ldots,m$ where x_i is the position of the ith gene ($x_1 \le \cdots \le x_m$)and y_i is the corresponding log-ratio. Let $\omega \ge 0$ (the neighborhood size) be a given integer. Define $\{\hat{f}_k\}$ to be the running mean of the data $\{y_k\}$ using a symmetric nearest neighborhood and $2w+1$ neighbors. Specifically, for genes not close to the boundary ($k = w+1,\ldots,n-k$) we define

$$\hat{f}_k = (y_{k-w} + y_{k-w+1} + \cdots + y_{k+w})/(2w+1).$$

For the remaning genes ($k = 1,\ldots,w$ and $k = n-w+1,\ldots,n$), we let $s = \max(1,\ k-w)$ and $t = \min(n,\ k+w)$, and define

$$\hat{f}_k = (y_s + y_{s+1} + \cdots + y_t)/(t-s+1).$$

The edge preserving smoother

As before we have a set of points (x_i, y_i), $i = 1, \ldots, m$, ordered such that $x_1 \le \cdots \le x_m$. An edge preserving smoother seeks to determine a sequence $\{\hat{f}_k\}$ that approximates the sequence $\{y_k\}$ well and contains as few jumps as possible (by a jump in the sequence $\{\hat{f}_k\}$ we mean a pair of consecutive elements $\hat{f}_i$ and $\hat{f}_{i+1}$ that satisfies $\hat{f}_i \neq \hat{f}_{i+1}$).

The edge preserving smoother that is implemented in CGH-Explorer is known as a Potts filter (see e.g. Winkler and Liebscher (2002)). Given a penalty parameter $\lambda > 0$ the values $\hat{f}_k$ are found by minimization over the scalars $f_1, \ldots, f_m$ of the penalized least squares criterion

$$H(f_1, \ldots, f_m) = \sum_{i=1}^{m} (y_i - f_i)^2 \; + \; \lambda \cdot \#\{i \,|\, f_i \neq f_{i+1}\}.$$

The first term on the right-hand side is a goodness-of-fit term. The last term on the right-hand side is a regularization term that penalizes for jumps in the function values. It equals a (nonnegative) constant λ times the number of incides i for which $f_i \neq f_{i+1}$. Thus a constant function (for which $f_1 = \cdots = f_m$) has zero penalty, whereas a function that changes value in every point has maximal penalty. The magnitude of λ controls the trade-off between goodness-of-fit and smoothness (defined here as the number of jumps).

The solution to the above optimization problem can be found using dynamic programming; we skip the details here.

Appendix B

The ACE algorithm

Here, we discuss the technical details of the Analysis of Copy Errors (ACE) algorithm. A more detailed account of ACE, including further developments and refinements of the method, will be submitted for publication in the near future. The discussion below pertains only to the current implementation of ACE in CGH-Explorer.

Basic concepts

The purpose of ACE is to search for copy errors in array CGH data. The ACE algorithm may be applied to a single array or simultaneously to a collection of arrays. In the latter case, the statistical analysis will be based on properties derived from the whole set of arrays. Suppose there are data from $N \geq 1$ arrays, each with M genes (clones). Each array corresponds to a sample or individual. For each array the input to the algorithm is a set of suitably transformed (see below) copy number ratios (in the following referred to as responses):

$$y_{jk},\ \ j = 1,\ldots,m,\ \ k = 1,\ldots,n_j$$

where j corresponds to the *j*th chromosome arm and k corresponds to the *k*th gene (clone) on the *j*th chromosome arm (note that $M = n_1 + \cdots + n_m$). The transformation should be chosen to make the distribution of the responses approximately normal. A common choice is to let the responses be the logarithm (in base 2) of the copy number ratios.

In the following, we assume that the responses y_{ijk} have been centered, so that the expected response for normal DNA is zero (here and below, we refer to DNA with normal copy numbers as normal DNA[1]). When this condition is not met, approximate

[1] Copy numbers vary slightly even among healthy individuals in a population. However, it is not common to have array CGH data of normal DNA from all individuals in a study, and most array CGH studies seem to ignore copy number polymorphisms.

centering can often be achieved by subtracting from each response on an array the average of all responses on the array[2].

Consider the problem of identifying segments of DNA that deviates from normal DNA. The approach used by ACE is first to identify all potentially interesting segments (defined here as genomic regions for which the responses are dominantly positive or dominantly negative). Properties of each segment (their length and height; see below) are then compared with those of segments derived from normal DNA. Clearly, we need some knowledge of normal DNA to perform a comparison like this. Such knowledge may be derived from several sources, including normal DNA from the same individual and normal DNA from another individual (or several other individuals). These alternatives are useful when data for normal DNA samples are available, but they cannot be applied when such data are unavailable. In ACE, on the other hand, only data from the DNA samples that are included in the study are required[3] for the analysis. Hence, ACE may be used even in situations where normal DNA is unavailable.

The formal statistical computations in ACE are based on the assumption that normal DNA responses are independent and identically distributed as $N(0,\sigma^2)$. Note, however, that violations of this assumption mainly affects the computation of the positive false discovery rate and not the classification rule applied to the data to distinguish aberrant DNA from normal DNA. The above assumption reflects the belief that response variability in normal DNA is due to the measurement process and the assumption that noise contributions for different genes should be independent (this may not be exactly true because of e.g. spatial effects on an array, but that will be ignored here). The unknown variance parameter σ^2 is estimated from the DNA samples in the study using only regions that are likely to be normal; details are provided further down.

ACE performs a series of tests to detect gene events, a gene event being that a particular gene in a particular sample (individual) is subject to loss or gain. There are NM null hypotheses, each stating that a particular gene in a particular sample has normal copy

[2] If centering has not been performed on the data prior to import into CGH-Explorer, you may perform it in CGH-Explorer by selecting the data set and choosing **Data | Mean-center arrays**.
[3] ACE can easily be extended to utilize data from normal DNA samples when such data are available. However, this is not implemented in the current version of CGH-Explorer.

number (hence follows a $N(0,\sigma^2)$ distribution). The steps involved in the ACE algorithm are described next.

Step 1: Segmentation

The purpose of this step is to identify in each array all possible candidate loss and gain regions. Specifically, we seek a subdivision of the genome into a minimal number of regions each of which are dominantly negative or dominantly positive with respect to the responses. This will be made precise below.

Consider one array and let $y_1,\ldots,y_n$ be the responses for all genes on a chromosome arm, listed in the same order as the genes are positioned along the chromosome. We assume that each gene has been assigned a unique position, which may for example be the position on the chromosome of the first nucleotide of the gene.

As a first step we want to assign each gene to one of two groups, depending on whether it is a candidate for being part of a loss region or part of a gain region. Note that, at this stage of the analysis, we do not seek to determine the strength of evidence in favor of loss or gain, and accordingly there is no third group corresponding to those genes that are neither candidates for being part of a loss or a gain region. In order to make the assignment of each gene to one of the two groups, we consider the gene as well as a small neighborhood around the gene. The method to be described here does not utilize the actual genomic distance between neighboring genes, only the order of the genes along the chromosome[4]. Hence, in referring to a small neighborhood we only intend to imply that it is small with respect to the number of genes in the neighborhood. Let $\{\overline{y}_k\}$ denote the running mean of the data $\{y_k\}$ using a symmetric nearest neighborhood and $2w+1$ neighbors (in the version of ACE now implemented in CGH-Explorer, we use $w=2$). For genes far away from the boundary, $k=w+1,\ldots,n-k$, we have $\overline{y}_k=(y_{k-w}+y_{k-w+1}+\cdots+y_{k+w})/(2w+1)$. For genes close to the boundary, $k=1,\ldots,w$ and $k=n-w+1,\ldots,n$, we define $s=\max(1,\ k-w)$ and $t=\min(n,\ k+w)$, and let $\overline{y}_k=(y_s+y_{s+1}+\cdots+y_t)/(t-s+1)$.

[4] An interesting extension of the method described here would be to take into account the actual physical distance between neighboring genes. In that perspective, the currently implemented method essentially assumes a uniform distribution of genes along the chromosome.

We now return to the question of assigning each gene to one of two groups depending on whether it is a candidate for loss or a candidate for gain. Define a binary classification $\{\zeta_k\}$ of the genes, based on the signs of the running mean terms (with a small modification). Let $\xi_k = sign(\overline{y}_k)$, where $sign(y) = 1$ if $y \geq 0$ and $sign(y) = -1$ if $y < 0$. We now let $\zeta_k = \xi_k$, unless $\xi_{k-2} = \xi_{k-1} \neq \xi_k \neq \xi_{k+1} = \xi_{k+2}$, in which case $\zeta_k = \xi_{k-1}$. In words, ζ_k is the sign of $\overline{y}_k$, unless all four neighbors $\overline{y}_{k-2}, \overline{y}_{k-1}, \overline{y}_{k+1}, \overline{y}_{k+2}$ have the opposite sign, in which case ζ_k equals the sign of these four neighbors. This classification rule is basically a robust version of the rule that assigns a gene to one of the two groups based on the sign of a local average of the responses around the gene.

The binary classification $\{\zeta_k\}$ induces a partitioning $J_1 \cup J_2 \cup \cdots \cup J_R$ of the gene indices $\{1,2,\ldots,n\}$ into $R > 0$ sets of consecutive indices, such that $\zeta_k = \zeta_l$ for any $k,l \in J_r$ and $\zeta_k \neq \zeta_l$ for any $k \in J_r$ and $l \in J_{r+1}$. For example, if $\zeta_1 = \zeta_2 = \cdots \zeta_6 \neq \zeta_7$, then genes 1-6 form a segment and $J_1 = \{1,\ldots,6\}$, and if $\zeta_7 = \zeta_8 \neq \zeta_9$ then genes 7 and 8 form a segment and $J_2 = \{7,8\}$, and so on. A segment is called positive if $\zeta_k = 1$ for genes in the segment and negative if $\zeta_k = -1$ for genes in the segment.

Step 2: Feature extraction

We want to be able to distinguish between segments that are likely to arise from segmentation of normal DNA, and those that are not. In order to do this, we consider two properties of each segment: their length (L) and height (H). Specifically, for each segment we compute the pair (L,H), where $L > 0$ is the number of genes in the segment and $H \geq 0$ is the absolute value of the average of the responses of the genes in the segment.

Step 3: Obtain the null distribution of the (L,H)-pairs

Let $f(L,H|\sigma^2)$ denote the density of (L,H)-pairs under the null hypothesis that responses are independent and identically distributed as $N(0,\sigma^2)$. The exact form of $f(L,H|\sigma^2)$ will not be discussed here; however one may show that, to a good approximation we have $f(L,H|\sigma^2) \approx f_0(L,\sigma^{-1}H)$ for a suitable density f_0. That is, the length of segments does not depend on the null variance, and the height of segments scales as σ. Accordingly, we need only find the distribution of (L,H)-pairs for independent and identically distributed $N(0,1)$ responses in order to determine $f(L,H|\sigma^2)$ for all σ.

Suppose now that $\hat{f}_0(L,H)$ is an approximation to $f_0(L,H)$, found by Monte Carlo simulation using $N(0,1)$ responses. For any given $\sigma > 0$ we approximate the density $f(L,H|\sigma^2)$ by $\hat{f}(L,H|\sigma^2) = \hat{f}_0(L,\sigma^{-1}H)$. Figure 1 shows the approximation $\hat{f}_0(L,H)$ resulting from one particular Monte Carlo simulation (approximations converge reasonably fast as a function of the number of simulated (L,H)-pairs).

In practice, the null variance has to be determined from the data. Suppose $\hat{\sigma}^2$ is an estimate of σ^2 (see later section on how $\hat{\sigma}^2$ is defined). Then our approximation for the null distribution of the (L,H)-pairs will be $\hat{f}(L,H) = f(L,H|\hat{\sigma}^2)$.

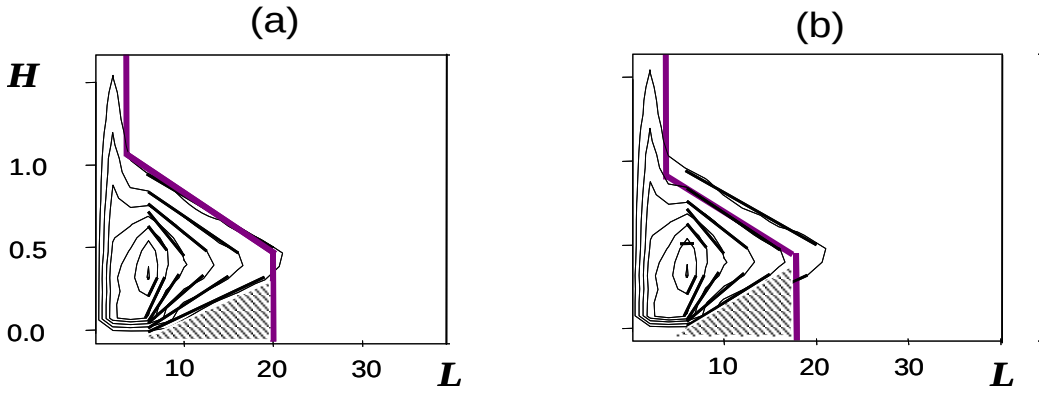


Figure 1: Contour plots showing the empirical distribution of (L,H)-pairs for simulated null data with variance $\sigma^2 = 1$. (a) The border between the rejection region (right region) and acceptance region (left region) for one particular significance level $\alpha(\lambda)$ is shown as a broken line; (b) The border between the rejection region (right region) and acceptance region (left region) for a different significance level $\alpha(\lambda') > \alpha(\lambda)$ is shown as a broken line.

Step 4: Find significant genes

A gene is called significant in ACE if it belongs to a segment for which the density $\hat{f}(L,H)$ is below a given threshold. This is analogous to how significance is defined in, e.g., a common t-test. However, in ACE there are two situations that are treated as exceptions from the above rule. First, genes in very short segments (say $L \leq L_{\min}$) are not called significant even if the density is below the treshold, since the possibility of outliers may have to be ruled out first. Second, genes in segments of moderate length and low height (the shaded areas in Figure 1) are not called significant, even though the null density is low in this part of the (L,H)-space. The reason for this is that we don't expect the density in this part of the (L,H)-space to be higher under the alternative hypothesis

(that would essentially imply that the variance is smaller under the alternative hypothesis than under the null hypothesis, and we have seen no empirical evidence in favor of this). Define $\hat{f}^*(L,H) = \max\{\hat{f}(L,h)) : h \geq H\}$. For $\lambda > 0$ define the acceptance region

$$\Omega_\lambda = \left\{(L,H) : L \leq L_{\min} \ \vee \ \hat{f}^*(L,H) > \lambda\right\}$$

and the corresponding rejection region

$$\bar{\Omega}_\lambda = \left\{(L,H) : L > L_{\min} \wedge \hat{f}^*(L,H) \leq \lambda\right\}$$

In the version of ACE currently implemented in CGH-Explorer we have $L_{\min} = 0$, i.e. there are no lower limit on the length of the segment to which a significant gene belongs.

For a sequence of λ values $0 < \lambda_1 < \cdots < \lambda_D$ we compute the proportion of genes that belong to (L,H)-pairs that fall in $\bar{\Omega}_\lambda$ under the null hypothesis. We refer to this proportion as a significance level and denote it by $\alpha(\lambda)$. We also compute the number of *observed* genes S_λ that belong to (L,H)-points in $\bar{\Omega}_\lambda$ (we refer to this number as the number of significant genes for the given level λ).

Step 5: Estimate the positive false discovery rate

Recall that M is the total number of genes on the arrays and N is the number of arrays. Hence, the total nu$V$mber of hypotheses is NM which typically is a very large number. Suppose each test is performed at level $\alpha(\lambda)$ and that the tests are independent (this is only approximately true in ACE). The expected number of false positives is then approximately equal to $NM\alpha(\lambda)$, assuming that the number of true positives is relatively small compared to the total number of hypotheses. In order to determine the appropriate level $\alpha(\lambda)$ to use in the tests, we follow Storey and Tibshirani (2003) and consider the positive false discovery rate (pFDR), defined as the conditional expectation of V/R when $R > 0$, where V denotes the number of false positives and R denotes the total number of positives (rejected hypotheses). The following estimate for pFDR is used in ACE:

$$pFDR(\lambda) = \frac{NM\alpha(\lambda)}{S_\lambda}$$

where S_λ as before denotes the number of genes that belong to segments for which (L,H) is in $\bar{\Omega}_\lambda$. The ratio $NM\alpha(\lambda)/S_\lambda$ is thus approximately equal to the expected proportion of false positives among all positives.

Step 6: Report genes

The final step in ACE is to report a list of genes assessed at a particular significance level to have altered copy number. In CGH-Explorer, the significance level is selected by the user from a list of possible levels, on the basis of the corresponding estimate of the positive false discovery rate for each level. For the chosen level, we find (as explained above) the segments for which the pair (L,H) belongs to $\bar{\Omega}_\lambda$ and call the genes in these segments significant. However, the precise start and end points of a segment may require adjustment because of the way segments are defined using local averages. Consider a significant segment, i.e. a segment for which the pair (L,H) belongs to $\bar{\Omega}_\lambda$. Let $z_1,\ldots,z_q$ be the associated responses. Define $K = \min(16, \lfloor L/2 \rfloor)$ and consider all possible subdivisions of the segment into three subsegments of length L_1, L_2 and L_3 respectively, where $0 \le L_1, L_3 \le K$ and $L_2 = L - L_1 - L_3$. For each subdivision (characterized by the pair (L_1, L_3)) compute the sum of squares

$$E(L_1, L_3) = \sum_{i=1}^{L_1} z_i^2 + \sum_{i=L_1+1}^{L-L_2} (z_i - H)^2 + \sum_{i=L-L_2+1}^{L} z_i^2$$

and find a subdivision (L_1^*, L_3^*) that minimizes the above criterion. The genes associated with the responses z_i, $i = L_1 + 1,\ldots,L - L_2$ are then reported as genes with loss if the average of the responses is negative, and as genes with gain if the average of the responses is positive.

Estimation of the variance parameter σ^2

We need to estimate the null variance σ^2 of the log copy number ratios. ACE performs this task by first identifying (in all the given DNA samples) regions of DNA that are likely to be normal and then calculating the variance based on data from those regions. Let $\tilde{\sigma}^2$ be a pilot estimate of σ^2, obtained by computing the variance of the responses of the genes that belong to segments for which $L \le 10$. Let $\tilde{f}(L,H) = f(L,H|\tilde{\sigma}^2)$ and choose $\tilde{\lambda} > 0$ such that roughly 50% of the total NM gene measurements belong to (L,H)-pairs that fall in $\tilde{\Omega} = \{(L,H) : \tilde{f}(L,H) > \tilde{\lambda}\}$. Finally, compute the variance of the responses of the genes that belong to (L,H)-pairs that fall in $\tilde{\Omega}$, and use the result as the desired variance estimate $\hat{\sigma}^2$.

Appendix C

Dealing with memory problems

You may experience memory problems in CGH-Explorer if you import large amounts of data, or if you perform many memory intensive operations (ACE in particular is memory intensive). Memory problems may in the worst case lead to a program crash unless you take precautions. There are two important factors related to memory usage:

- The total amount of memory that is available to CGH-Explorer. This is determined at program launch time by the operating system and cannot be changed during a program session.

- The proportion of the available memory that have been used so far. This proportion increases every time you import a new data set, make a plot or perform an ACE analysis. Note that closing a window in CGH-Explorer (e.g. a plot window or a data selection window) does not free any memory, since closing a window simply removes the window temporarily from sight. Closed windows can always be reopened using the **Window** menu.

Use the toolbar to see if you are in danger of experiencing a memory problem. The toolbar tells you how much memory is available for CGH-Explorer, and how much of that memory is used so far:

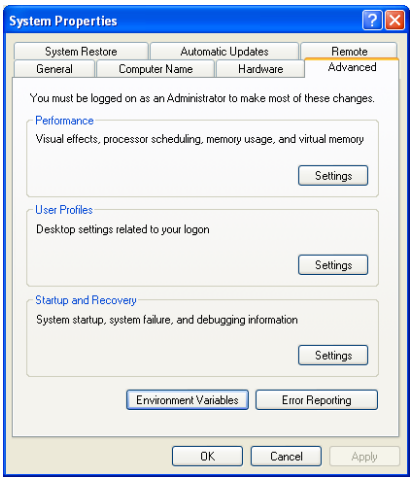
Memory used so far by CGH-Explorer

Total amount of memory available to CGH-Explorer

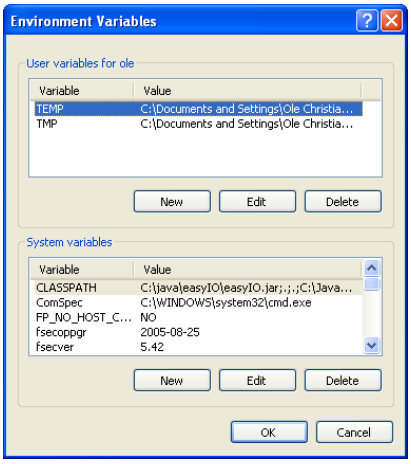
Suppose the proportion of memory used so far exceeds some threshold (say 75% of available memory) and you want to import new data or perform further analyses. You may then want to consider terminating the current program session and increasing the total amount of memory available to CGH-Explorer. To do this you must define an environmental variable called JETVMPROP. Below we explain how you do this in Windows XP:

1. Open the **Control Panel** in Windows and start the program called **System** (note that the Control Panel and/or the program System may be hidden from view depending on your Windows preferences – in that case you must change Folder Options).

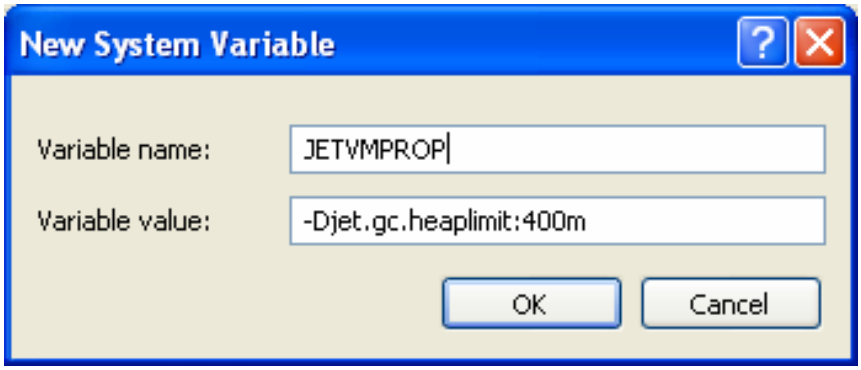
2. Click on the **Advanced** tab. The result so far should be as follows:



3. Now click on the button **Environmental Variables** to show a window like this:



4. The bottom list of system variables consists of two columns: the left column is the name of the system variable and the right column is the value of the variable. Make sure that no variable with the name JETVMPROP is defined (if it is, you probably have another program installed that depends on this variable. What to do in that case depends on the circumstances and is outside the scope of this manual). Click on the button **New** below the list of system variables and fill in exactly as follows (the number 400 means that we extend the memory available to CGH-Explorer to a total of 400Mb; you may want to use another number here depending on your needs and the amount of available memory on your computer):



5. Click on OK in all the dialog windows to finish.

REFERENCES

Storey JD and Tibshirani R (2003). Statistical significance for genomewide studies. *PNAS* 100, 9440-9445.

Winkler G and Liebscher V (2002). Smoothers for Discontinuous Signals. *J. Nonpar. Statist.* 14, 203-222.